

1 引言

近年来，人们逐渐从信息化时代迈向了数据时代，各种数据爆炸式地增长，数据消费也在日益增多，大量的信息、知识和利润隐藏在这些数据中。如何更有效地利用这些数据，已经成为这个时代下人们共同探索的问题之一。

在这次大作业中，我将对 Adult 数据集进行全面的分析：首先探索数据集中各特征的分布信息；再划分数据集，尝试多种分类模型；最后比较这些模型在 Adult 数据集上的预测结果（分析代码均基于 Python 语言，相关工具和库包可参见附录 A.1）¹。

2 探索 Adult 数据集

2.1 Adult 数据集的基本信息

Adult 数据集 [1] 也称人口普查收入（Census Income）数据集，来源于美国 1994 年的人口普查数据库，可以作为二分类数据集，用来预测居民年收入是否超过 50K\$，其基本信息可参见表 1。

表 1: Adult 数据集的基本信息

属性	值	属性	值
数据集特征	多变量	相关应用	分类
实例数	48842	捐赠日期	1996.5.1
领域	社会	是否有缺失值	有
属性特征	类别型或整数	官网访问次数	1188850
属性数目	14		

Adult 数据集的每个实例包含 14 个属性，其含义、数据类型、取值范围等基本信息见表 2。

表 2: Adult 数据集的基本信息

特征名	含义	数据类型	类别数
age	年龄	整数	-
workclass	工作类型	类别型	8
fnlwgt	序号	整数	-
education	教育程度	类别型	16
education-num	受教育时间	整数	-
marital-status	婚姻状况	类别型	7
occupation	职业	类别型	14
relationship	家庭关系	类别型	6
race	种族	类别型	5
sex	性别	类别型	2
capital-gain	资本收益	整数	-
capital-loss	资本损失	整数	-
hours-per-week	每周工作小时数	整数	-
native-country	原籍	类别型	41

¹本次大作业以及以往小作业的代码可参见我的 github 仓库：<https://github.com/shinshiner/CS245-Data-Science>

2.2 数据预处理

我首先使用 `pandas` 库读取 `Adult` 数据集，将其存储为 `pandas` 库中的 `DataFrame` 格式，随机打印出其中几个实例，对该数据集进行初步的观察，结果如下。

1	age	work_class	fnlwgt	education	education_num	marital_status
2	24	Private	269799	Assoc-voc	11	Never-married
3	35	?	169809	Bachelors	13	Married-civ-spouse
4	51	Private	257126	10th	6	Married-civ-spouse
5	72	Private	107814	Masters	14	Never-married
6	33	Private	205950	HS-grad	9	Never-married
7						
8	occupation		relationship	race	sex	capital_gain
9	Exec-managerial		Not-in-family	White	Male	0
10	?		Husband	White	Male	0
11	Craft-repair		Husband	White	Male	0
12	Prof-specialty		Not-in-family	White	Male	2329
13	Other-service		Own-child	White	Male	0
14						
15	capital_loss	hours_per_week	native_country	income		
16	0	40	United-States	<=50K.		
17	0	20	United-States	<=50K		
18	0	40	United-States	<=50K.		
19	0	60	United-States	<=50K		
20	0	40	United-States	<=50K		

从以上的初步观察可以得知，`Adult` 数据集存在数据缺失的情况（如第 3 行和 10 行的“？”），我对整个数据集进行统计后，发现数据集中共有 3620 个实例存在缺失值，而其中 2799 个实例的缺失值多于 1 个（表 3）。同时，我发现分类目标（`income`）的部分值存在歧义，“<=50K.”与“<=50K”属于同类，却被赋上不同标签，在后续预处理过程中（3.3.3 小节）我会进行处理。

表 3: `Adult` 数据集缺失值分布

	无缺失值	缺失 1 个特征	缺失 2 个特征	缺失 3 个特征
实例数	45222	821	2753	46

考虑到数据集中存在缺失值的实例数较少（仅占总数的 7.41%），且缺失的均为类别型变量，若用一般的方式填补会带来较大的偏差，因此我选择将这些实例直接删除，清理缺失值后的 `Adult` 数据集包含 45222 个实例，虽然由于删除数据导致 `workclass` 特征减少了一类（`Never-worked`），但相应的实例只有 10 个，可以忽略不计。

2.3 `Adult` 数据集中各特征的分布

对 `Adult` 数据集进行检查和清理后，我开始探索 `Adult` 数据集中各特征的分布，我将数值型特征和类别型特征分别处理：对于数值型特征，我主要关注其数字特征（如均值，方差等）及分布密度；对于类别型特征，我主要关注其具体的分布情况。

2.3.1 数值型特征的分布

Adult 数据集中的数值型特征为: age, fnlwgt, education-num, capital-gain, capital-loss 以及 hours-per-week。我首先统计其均值、标准差等数字特征, 相应结果如表 4 所示。

表 4: Adult 数据集数值型特征的数字特征

特征名	均值	标准差	最大值	最小值	上四分位数	下四分位数
age	38.548	13.218	90.000	17.000	47.000	28.000
fnlwgt	18976.470	10563.920	1490400.000	13492.000	237926.000	117388.200
education-num	10.118	2.553	16.000	1.000	13.000	9.000
capital-gain	1101.430	7506.430	99999.000	0.000	0.000	0.000
capital-loss	88.595	404.956	4356.000	0.000	0.000	0.000
hours-per-week	40.938	12.008	99.000	1.000	45.000	40.000

上表的数据大致反映了各特征的分布情况, 为更加直观地探索各数值型特征的分布趋势, 我作出了相应的概率核密度分布图 (高斯核), 见图 1。

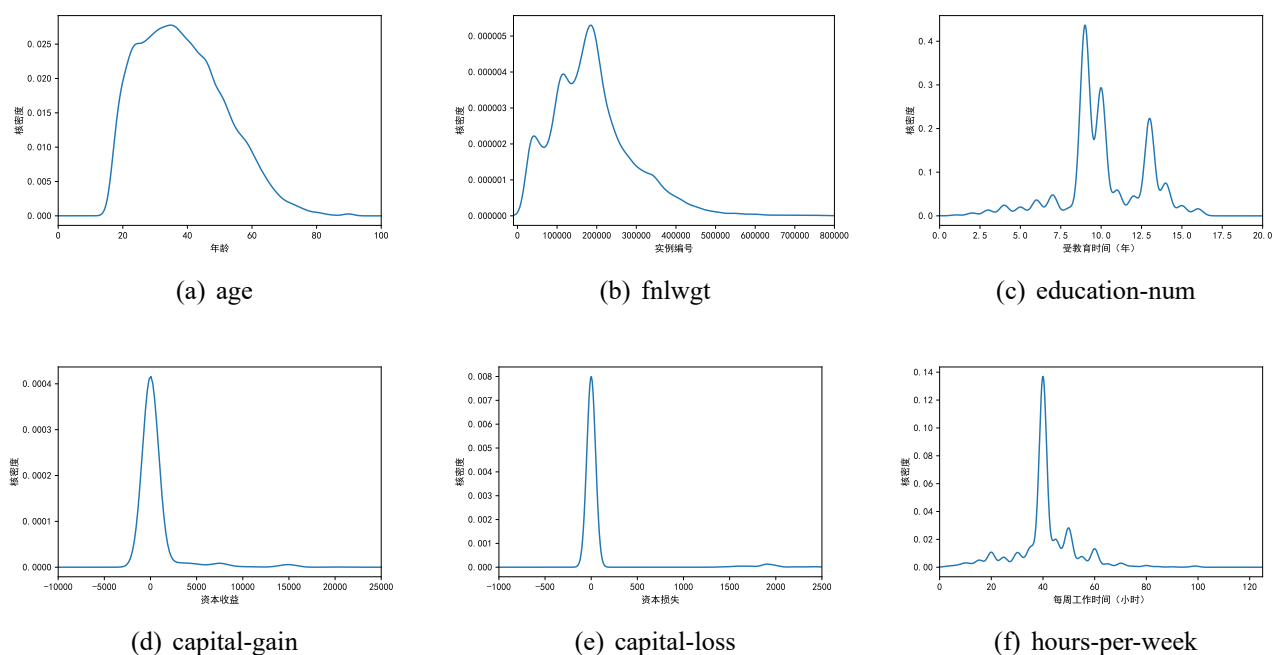


图 1: Adult 数据集数值型特征概率核密度分布

容易看出, Adult 数据集的 6 个数值型特征接近于正态分布。通过进一步的观察, 我发现 capital-gain 和 capital-loss 这两个特征的大部分取值均分布在 0 附近, 仅通过概率核密度图无法了解两特征其余取值的分布情况。为更精确、详细地探索其分布, 我做出了两特征对数值 ($\log(x+1)$) 下相应的直方图 (图 2)。

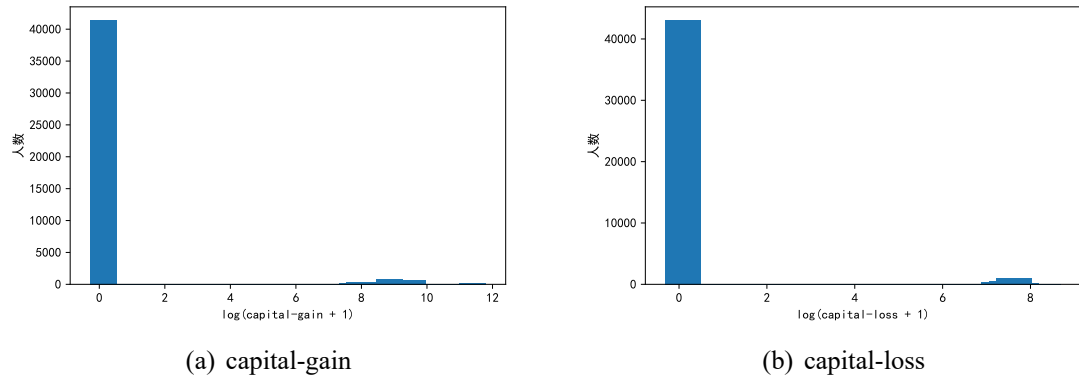


图 2: capital-in 和 capital-out 的分布直方图

2.3.2 类别型特征的分布

Adult 数据集中的类别型特征包含: workclass, education, marital-status, occupation, relationship, race, sex 以及 native-country。我将其分布表示为条形图或饼图 (图 3, 4)。各特征中包含的详细类名已记录于附录 A.2.2 中。

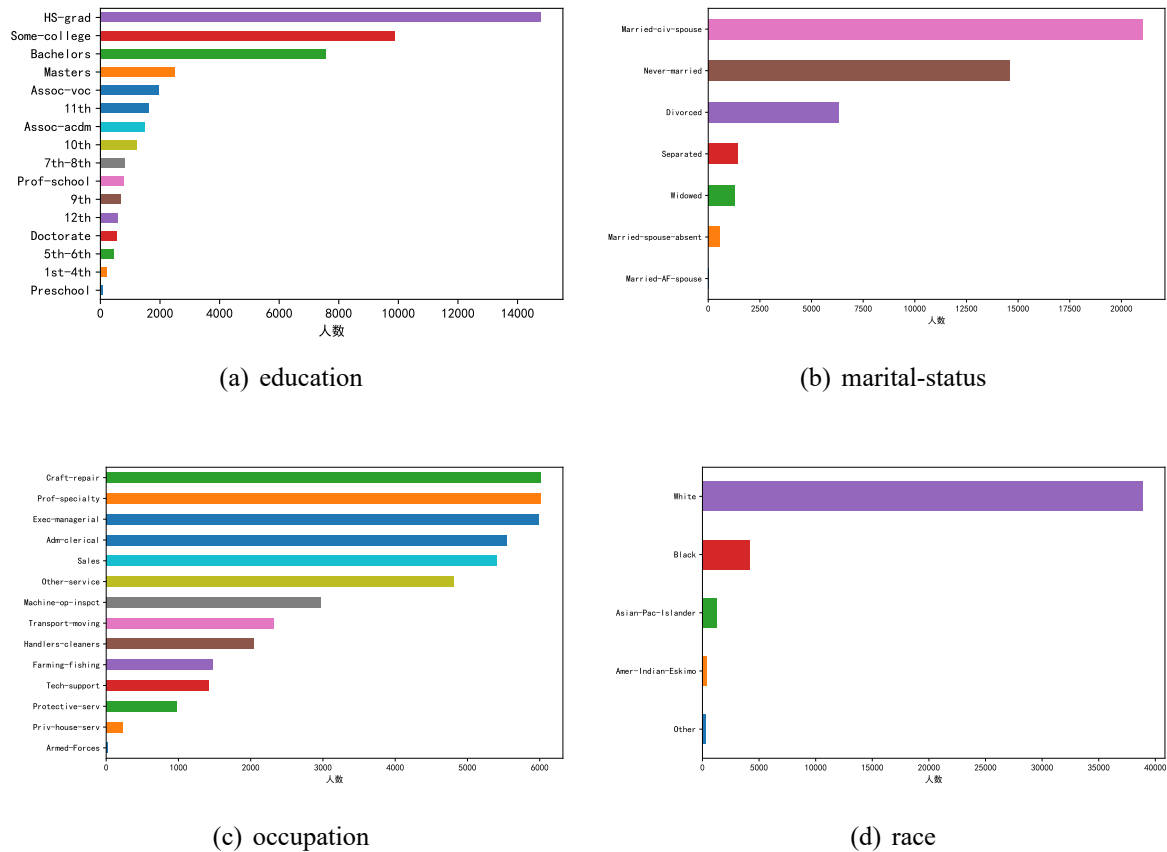


图 3: Adult 数据集类别型特征分布条形图

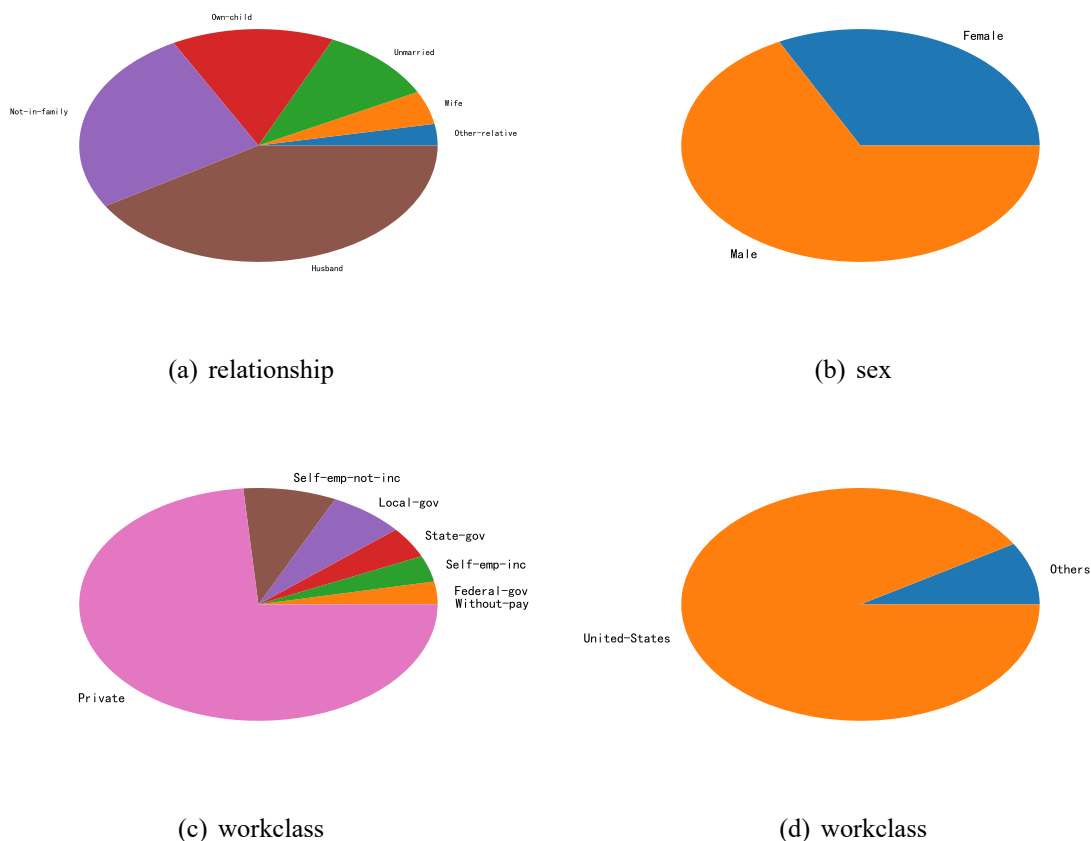


图 4: Adult 数据集类别型特征分布饼图（由于 native-country 包含的类数较多，因此除 United-States 外国籍的分布见附录 A.2.1）

2.4 Adult 数据集中各特征的相关性

探索完 Adult 数据集中各特征的分布后，我开始探索特征之间的相关性，

3 划分数据集并构造分类模型

探索完 Adult 数据集中各特征的分布和相关性后，我开始对其进行训练集和测试集的划分并构造一系列分类模型。

3.1 划分数据集

对数据集的划分一般有两种方法：一是直接按照一定的比例将数据划分为训练集和测试集（需保证训练集和测试集中的类分布大致相同）；二是使用分层交叉验证，将数据随机等分为 k 个不相交子集，执行 k 次训练与测试，根据 k 次迭代的平均表现评价模型的性能。

本次作业中，为使评价结果更加精确，我主要使用分层交叉验证方法划分数据集，只在训练基线分类模型（baseline）时使用直接划分训练集和测试集的方法。

3.2 构造分类模型

在机器学习领域，用于分类的算法种类繁多，基本的分类算法包括了逻辑回归（Logistic Regression）、K 近邻（KNN）、决策树、支持向量机（SVM）以及多层感知机（MLP）等。考虑到

Adult 数据集的特征维数并不高，且分类目标简单（二分类），我在本次作业中选择使用决策树、SVM 以及 MLP 三种分类模型（图 5）。除使用单独模型进行分类外，我尝试应用了模型集成的方法以提高相应的分类效果。

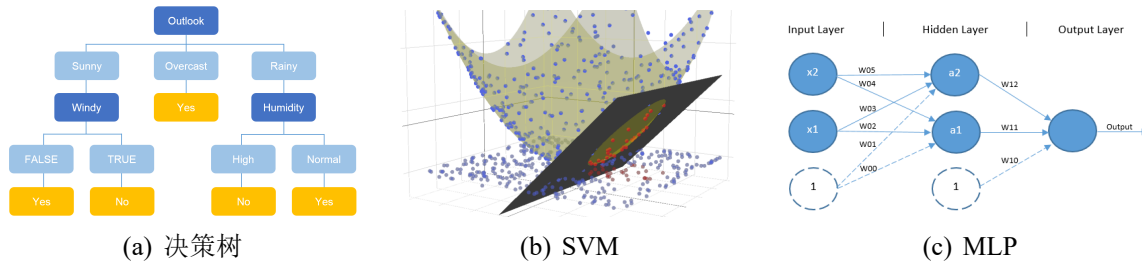


图 5: 决策树、SVM 和 MLP 模型的直观表示

3.3 数据预处理

在进行正式的分类之前，我对 Adult 数据集的特征和分类目标进行了一些预处理，以方便分类模型的训练。

3.3.1 Z-score 标准化（规范化）

一般地，Z-score 标准化有如下形式：

$$y = \frac{x - \mu}{\sigma} \quad (1)$$

其中 μ 和 σ 分别代表原数据的均值和标准差。

对于分类问题的数值型特征，经 Z-score 标准化后符合标准正态分布，即 $N(0, 1)$ ，可以有效避免因数值过大导致的模型偏差，并能够加快模型的学习速率，这些优势在 SVM 和 MLP 等模型中表现得更加明显。

除此之外，2.3.1 小节的结果表明，Adult 数据集里的数值型特征大多近似服从正态分布，在这个条件下，Z-score 标准化能够取得更良好的效果。若特征的分布与正态分布相差较大，则 Z-score 标准化反而会破坏原数据的分布，造成额外的偏差。

3.3.2 向量化

Adult 数据集中存在 8 个类别型特征，且这些特征的类别之间并无大小关系，如性别的男女之间不存在大小的区别。为方便模型的训练，我将这些特征从字符串转化为 one-hot 向量。经过向量化处理后的数据，每个实例包含 104 维特征。

3.3.3 分类目标修正

Adult 数据集的分类目标为居民收入，分为两类： $\leq 50K\$$ 以及 $> 50K\$$ 。而数据集中有些实例的分类目标后多了“.”，如变为“ $\leq 50K.$ ”，直接使用原数据训练将导致分类目标变为 4 类。因此，我将分类目标转化为 -1 和 1，分别表示 $\leq 50K\$$ 和 $> 50K\$$ 。另外值得注意的一点是 Adult 数据集包含 34014 个负样本（ $\leq 50K$ ），而仅有 11208 个正样本。

3.4 分类性能评价标准

对于分类问题，评价分类性能的标准一般有 3 个：精确度（precision），召回率（recall）以及 f1-score。其在二分类问题中的定义如下。

假设二分类问题的结果为：

	正类	负类
分类正确	TP	FP
分类错误	FN	TN

则精确度、召回率和 f1-score 分别定义为

$$\begin{aligned}P &= \frac{TP}{TP + FP} \\R &= \frac{TP}{TP + FN} \\F1 &= \frac{2PR}{P + R}\end{aligned}\tag{2}$$

一般地，在大规模的数据集下，精确度和召回率会相互制约，出现一高一低的现象，而 f1-score 可以兼顾二者，有效减少因一者过大带来的误差。因此，在本次作业的实验中，我主要以 **f1-score** 作为分类性能的主要评价标准，精确度和召回率则在训练基线模型时作为参考标准。

4 各分类模型的预测结果比较

4.1 基线模型（baseline）

为方便之后的比较，我首先使用 sklearn [2] 中的默认参数，不使用标准化，构造了三个基线模型，以及一个空模型（按相等概率随机预测），参见表 5 和表 6。

表 5: 基线模型在 Adult 数据集上的性能（0.8-0.2 比例的训练集-测试集分割）

	精确度（precision）	召回率（recall）	f1-score	时间（秒）
决策树	0.81	0.81	0.81	0.25
SVM	0.76	0.83	0.76	1044.22
MLP	0.78	0.80	0.79	0.59
空模型	0.72	0.52	0.57	0.07

表 6: 基线模型在 Adult 数据集上的性能（5 折分层交叉验证）

	精确度（precision）	召回率（recall）	f1-score	时间（秒）
决策树	0.80	0.80	0.80	4.41
SVM	0.83	0.83	0.83	4491.21
MLP	0.81	0.83	0.80	11.82
空模型	0.72	0.50	0.56	0.19

4.2 决策树模型

在本节中，我将使用网格搜索（Grid Search）对决策树模型的分类效果进行评估（5 折交叉验证），搜索的参数及范围参见表 7。

表 7: 对决策树模型网格搜索的参数及范围

参数	含义	类型（范围）
criterion	特征选择的度量标准	gini, entropy
max_depth	树的最大深度	正整数
max_features	寻求最佳划分时 要考虑的特征数目	总特征数或其平方根 或其以 2 为底的对数值
presort	是否对数据预先排序	布尔值
splitter	结点划分策略	best, random

经过网格搜索后，我得到的最佳决策树模型的参数为：`{criterion: entropy, splitter: best, max_features: 总特征数, max_depth: 9, presort: True}`；该模型在测试集上的平均 f1-score 为 0.83。

对于经过 Z-score 标准化的 Adult 数据集，网格搜索的结果为：`{criterion: entropy, splitter: best, max_features: 总特征数, max_depth: 9, presort: True}`，测试集上的平均 f1-score 为 xxx。

最终通过网格搜索筛选出的最佳决策树模型可视化见图 6。

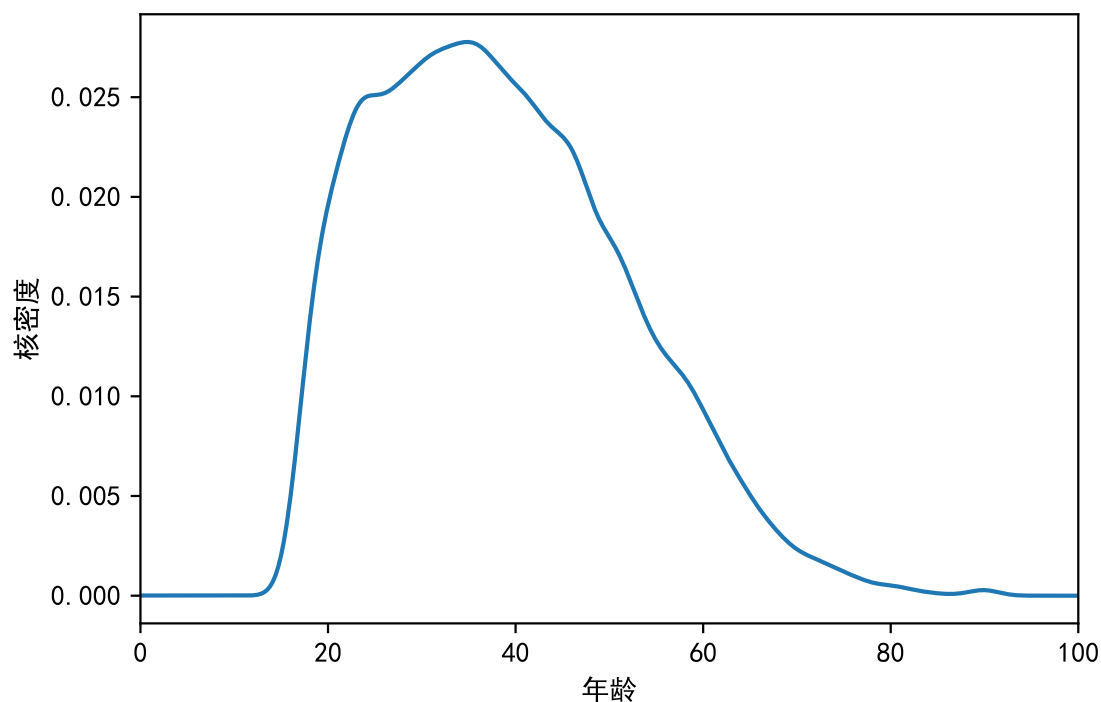


图 6: 最佳决策树模型可视化

4.3 SVM

从表 5 和表 6 来看，SVM 的二分类能力并没有完全展现出来，并且难以收敛。在这一节中，我将尝试改变 SVM 模型的多个关键参数，尽可能改善其表现（模型表现均基于 5 折交叉验证）。

4.3.1 核函数

原始的 SVM 属于线性分类器，核函数的引入将 SVM 的应用推广到了非线性数据上。不同的核函数所需要的计算量和性能均有较大差别，我在 Adult 数据集上尝试应用四种常用的核函数：rbf 核函数，多项式核函数（poly kernel），sigmoid 核函数以及线性核函数，各模型相应的表现见图 7 及图 ??。

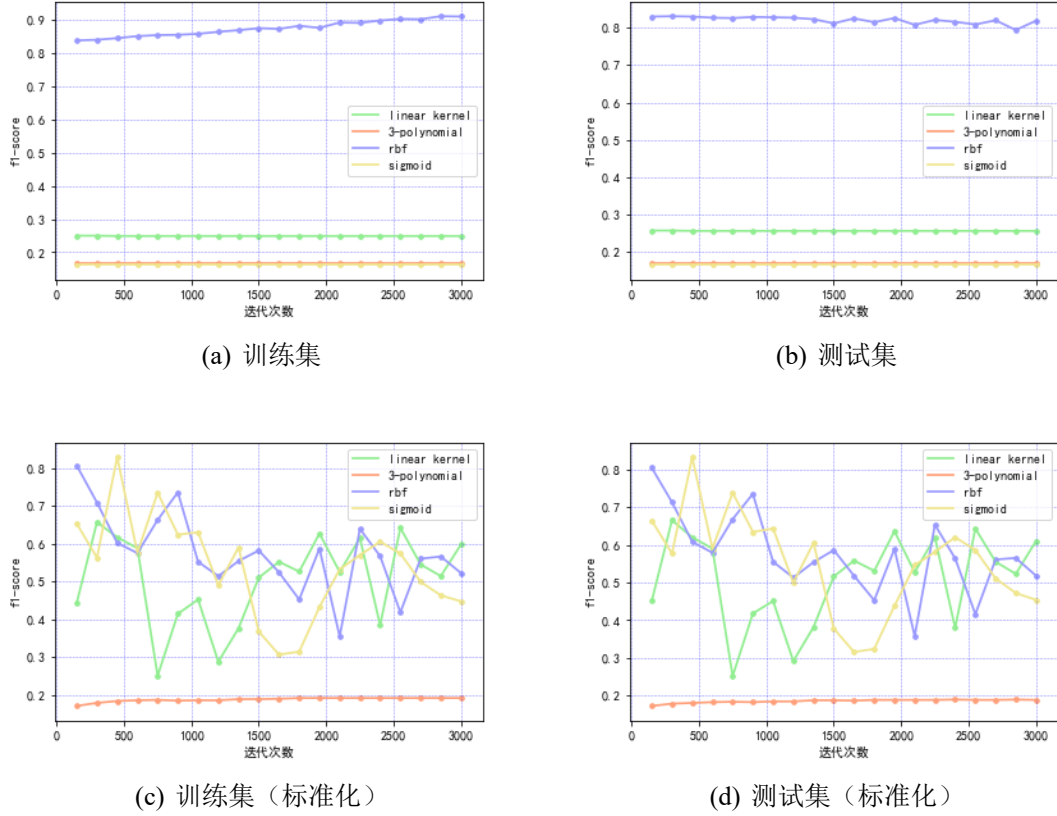


图 7: 不同核函数下 SVM 模型的分类表现

从上图中容易看出，对于未经 Z-score 标准化后的数据，SVM 模型在使用 rbf 核时分类效果较佳，而其余核函数的分类效果极差，甚至低于空模型；而对于 Z-score 标准化后的数据，linear 和 sigmoid 核函数的表现略有提高，但训练均极不稳定，向空模型的方向收敛。因此之后的实验均基于未 Z-score 标准化后的数据使用 rbf 核进行。

4.3.2 惩罚系数

SVM 在面对轻微的线性不可分数据时，可以通过引入惩罚系数 C ，将原有的优化目标

$$L = \frac{1}{2}w^T w \quad (3)$$

变为

$$L = \frac{1}{2}w^T w + C \sum_{n=1}^N \xi_n, \quad (4)$$

适当的惩罚系数可以极大地改善 SVM 的分类性能。我使用类似于网格搜索的方式，遍历了各数量级的 C 取值，相应的结果可参见图 8。

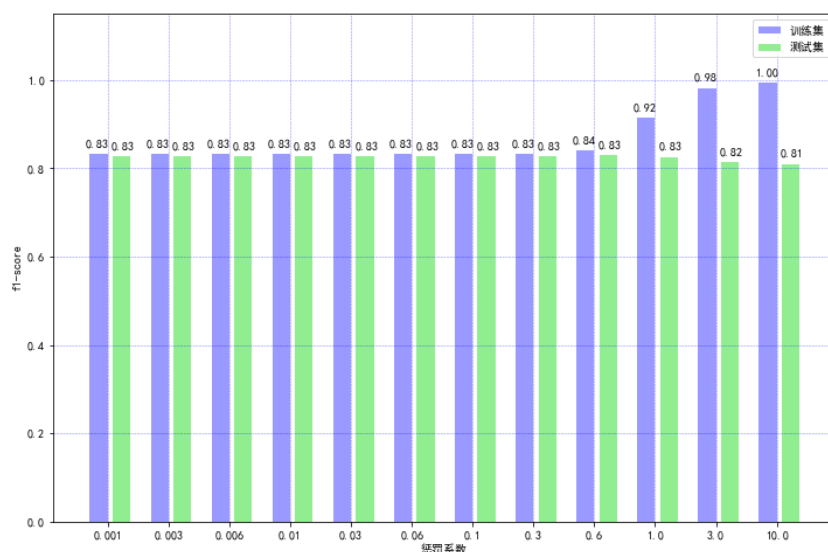


图 8: 不同惩罚系数下 SVM 模型的分类表现

上图结果表明, 过低的惩罚系数 C 对 SVM 在 Adult 数据集上的分类性能没有改善作用, 0.6-1.0 之间的 C 能够略微改善 SVM 的性能, 而过大的 C 会让 SVM 趋于过拟合, 泛化能力逐渐降低。

4.3.3 Gamma (γ)

rbf 核的数学表达形式为:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (5)$$

γ 是 rbf 核函数的一个重要参数, 在 sklearn 中, 其默认值为: $\frac{1}{\text{特征总维数}}$, 在 Adult 数据集上即为 $1/104 = 0.096$ 。我利用类似网格搜索的方式探索了各数量级下的 γ 对应的 SVM 模型的分类表现 (表 8)。

表 8: 不同数量级的 γ 对应的 SVM 模型的分类表现

γ	训练集 f1-score	测试集 f1-score
----------	--------------	--------------

4.3.4 小结

经过以上的探索, 我得到了 SVM 模型在 Adult 数据集分类上表现较佳的一组参数: {核函数: rbf, 惩罚参数: 1.0, γ : xxx, 是否进行 Z-score 标准化: 否}, 相应的 SVM 模型在测试集上的 f1-score 为: xxx。

4.4 MLP

从 MLP 基线模型 (表 5, 6) 的性能分析, MLP 的性能相比决策树模型稍逊。我认为较大的原因在于 MLP 模型的默认参数并不适合 Adult 数据集, 因此, 在本节中, 我将对 MLP 中不同的组件 (激活函数, 优化算法等) 对其在 Adult 数据集上分类性能的影响进行探究。

4.4.1 激活函数

激活函数作用于 MLP 隐藏层中的每个神经元上，为 MLP 引入非线性因素，若缺少激活函数，神经网络的表达能力将极其有限。因此，激活函数是整个 MLP 中重要的组件之一。常用的激活函数包括 relu [3]，tanh 以及 sigmoid 等，相应的预测表现见图 9。

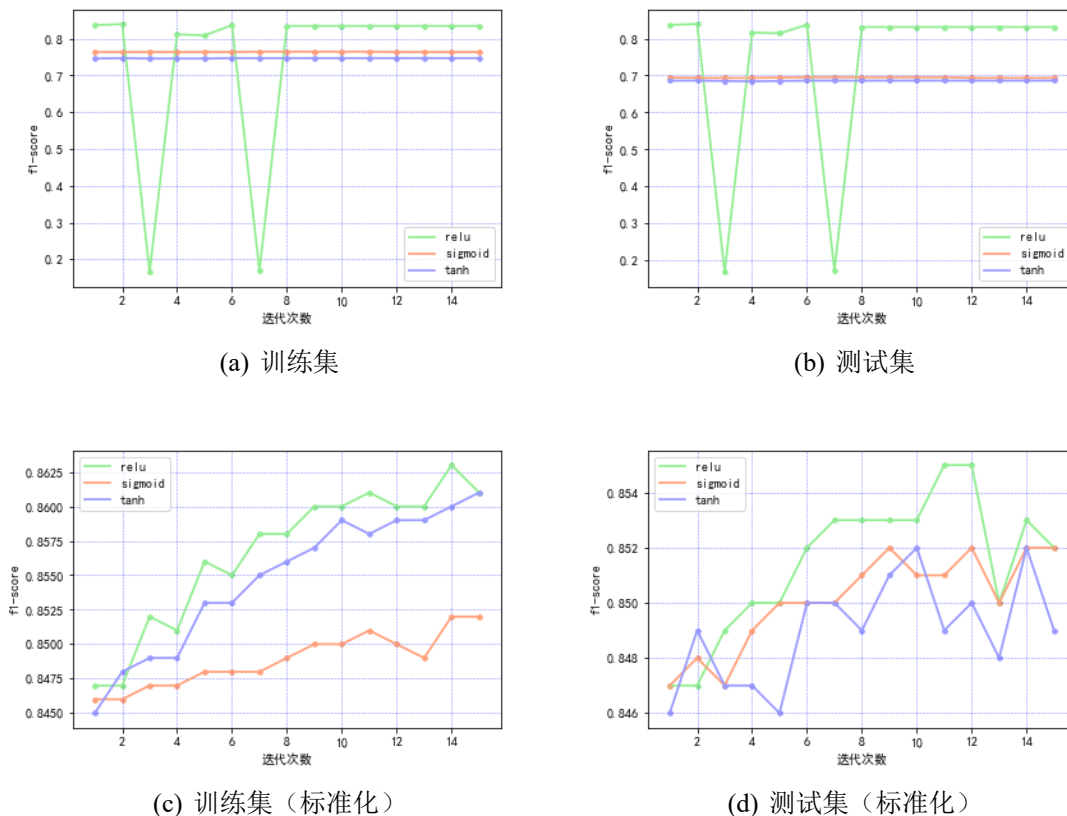


图 9: 不同激活函数下 MLP 模型的分类表现

上图结果表明，相比于 sigmoid 和 tanh，relu 激活函数在 Adult 数据集下的分类表现更佳。同时，MLP 在经过 Z-score 标准化后的 Adult 数据集上能取得更好的分类效果，不仅提高了整体的分类 f1-score，更有效地避免了使用 relu 激活函数时进入局部最小值的情况。因此，之后 MLP 部分的实验均基于 Z-score 标准化后的 Adult 数据集进行。

4.4.2 优化算法

优化算法作用于 MLP 更新参数时，对于同样的梯度分布，不同的优化算法的优化路径存在着极大的差别，图 10 形象地说明了这一点。因此，探究不同优化算法下 MLP 的分类性能是很有必要的，详细结果可参见图 13。

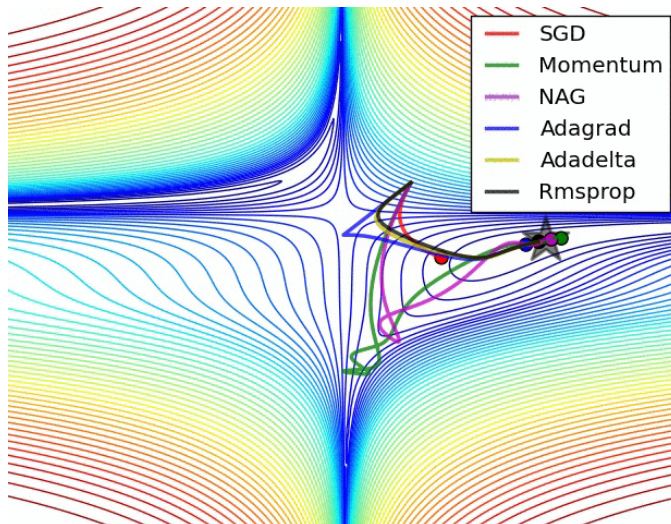


图 10: 不同优化算法的优化效果对比 [4]

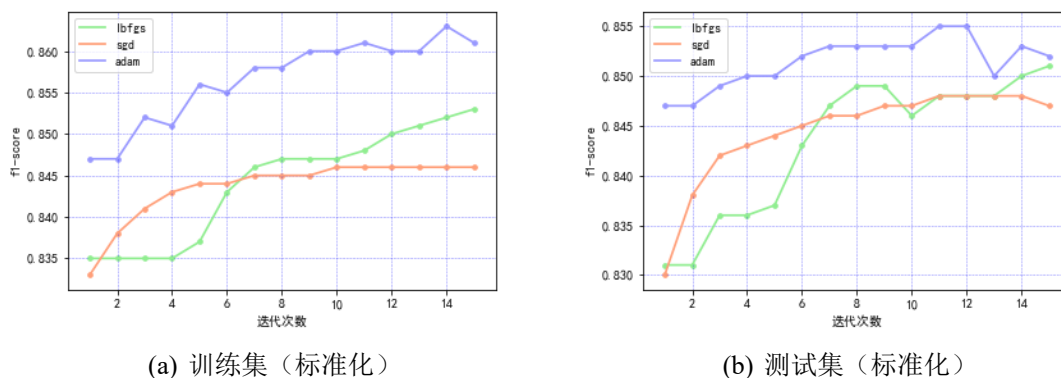


图 11: 不同优化算法下 MLP 模型的分类表现

根据上图结果，adam [5]、lbfgs 与 sg 三种优化算法均能使 MLP 在 Adult 数据集上取得较佳的分类表现，但 adam 的效果明显优于 lbfgs 与 sg，为探究其中的原因，我查阅了相关资料和原论文，认为图中反映出 adam 算法的巨大优势有着其坚实的理论依据。

作为最常用的优化算法，adam 能够利用较少的计算资源有效处理高噪声或稀疏的梯度分布，除此之外，其在优化的过程中能够通过计算梯度的一阶矩估计和二阶矩估计，为不同的参数设计独立的自适应性学习率。

4.4.3 学习率

在 MLP 的参数更新过程中，学习率决定了参数更新的速率，过大的学习率会导致模型进入局部最小值，而学习率过小则会减缓模型的学习速度。在本节中，我采用类似于 4.3.2 小节的方式，对基于 adam 优化算法的 MLP 模型在不同学习率下的分类效果进行探索（图 12），由于 adam 算法能够自适应地改变学习率，此处的学习率即指初始学习率。

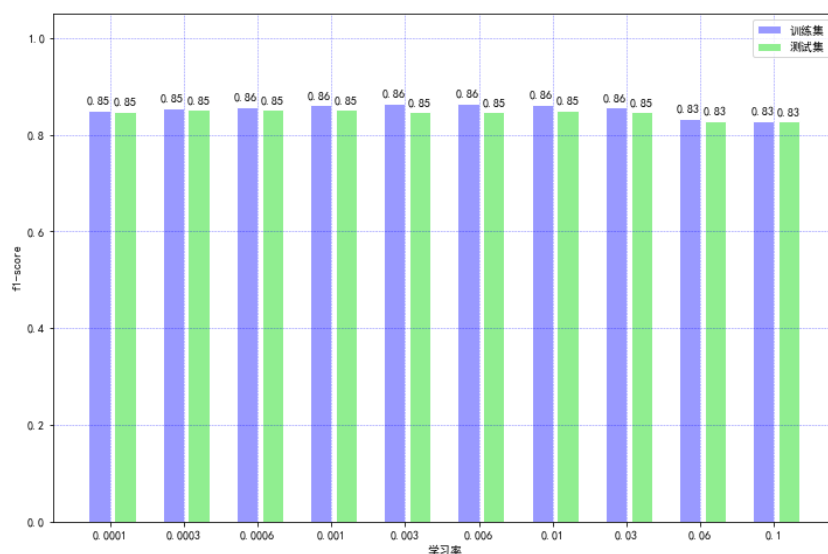


图 12: 不同初始学习率下 MLP 模型的分类表现

容易看出，除了过大的学习率有降低 MLP 性能的趋势外，MLP 在其余学习率下的表现并无太大差别，考虑到实际应用时的时间占用，我认为 0.03 是基于 adam 优化算法的 MLP 模型在 Adult 数据集分类问题下较佳的学习率。

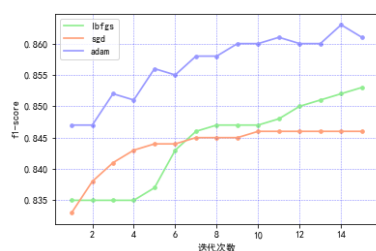
4.4.4 小结

经过以上的探索，我得到了 MLP 模型在 Adult 数据集分类上表现较佳的一组参数：{激活函数: relu, 优化算法: adam, 学习率: 0.03, 是否进行 Z-score 标准化: 是}，相应的 MLP 模型在测试集上的 f1-score 为: xxx。

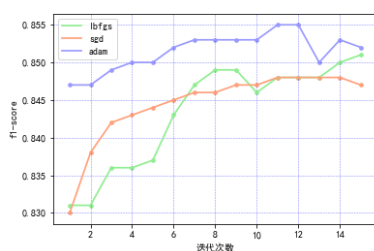
4.5 模型集成

从以上几节的结果分析，决策树模型在 Adult 数据集上的分类效果弱于 SVM 和 MLP。一般地，从一系列模型 M_1, M_2, \dots, M_k 创建组合模型 M^* ，可以有效提高原模型的效果。相比于 SVM 和 MLP，决策树模型训练时速度快、占用计算空间和资源较少，适合进行模型集成。

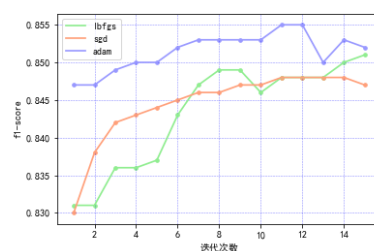
因此，在本节中，我使用 bagging 和 boosting 两种模型集成的方法尝试改善决策树模型的分类效果。结果参见图 ??。



(a) bagging



(b) boosting (SAMME)



(c) boosting (SAMME.R)

图 13: 集成后的决策树模型的分类表现

4.6 三种分类模型效果对比

通过 4 小节的实验，我对决策树、SVM 和 MLP 三种模型在 Adult 数据集上的分类效果有了大致的了解。在本节中，我将探索三种分类模型在不同条件下对 Adult 数据集的分类效果。

4.6.1 Z-score 标准化

4.6.2 PCA

4.6.3 最佳性能对比

根据 4 与 4.6 小节的实验结果，三种模型均存在一个最佳的 f1-score 值，一定程度上代表其在 Adult 数据集上分类能力的极限（由于模型的参数空间极其巨大，无法保证该值是否为全局最优，但在一定程度上能反映该模型的最佳性能），对比参见表 ??。

5 对 Adult 数据集的分析结论

Adult 数据集是一个中等规模，存在少部分缺失值的二分类数据集，共 48842 个实例，每个实例包含 14 个特征，其中 8 个为类别型特征，6 个为数值型特征。各类别型特征的分布差异较大，数值型特征的分布近似于正态分布。

我使用了决策树、SVM 和 MLP 三种分类模型对 Adult 数据集进行分类，对模型的参数进行微调后，xxx 的分类效果最佳，最高的 f1-score 可达到 xxx，而 xxx 模型所消耗的时间最少，平均不到 xxx 秒。

A 附录

A.1 作业中使用的工具及库包

本次作业我所使用的编程语言为 Python [6]，编辑环境以 jupyter notebook [7] 为主。作业中我使用的库包见表 9。

表 9: 本作业中使用的库包

库包名	用途
numpy [8]	处理数据，数值计算
pandas [9]	读取数据，绘图，处理数据
matplotlib [10]	绘图
scipy [11]	数值计算
scikit-learn [2]	分类模型的构造和运算

A.2 Adult 数据集特征分布补充资料

A.2.1 native-country 特征的详细分布

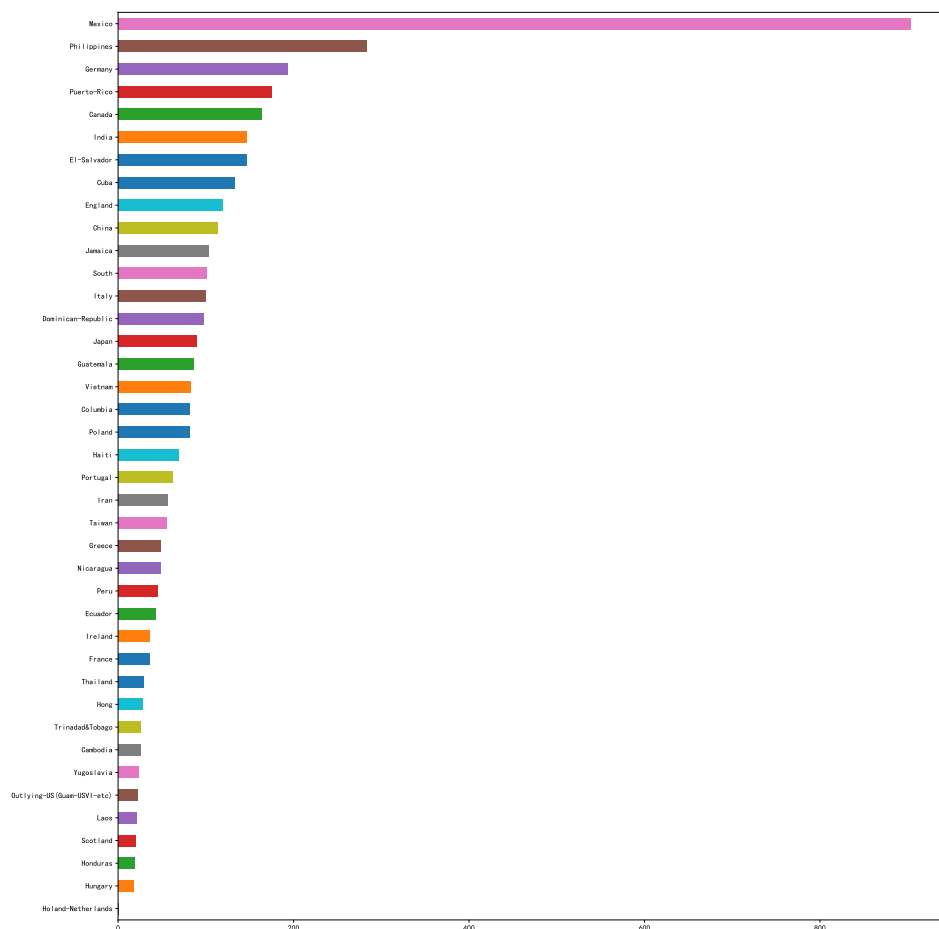


图 14: Adult 数据集 native-country 特征分布（除 United-States）

A.2.2 各类别型特征下的详细类名

1. **workclass** Private, Local-gov, Self-emp-not-inc, Federal-gov, State-gov, Self-emp-inc, Without-pay, Never-worked;
2. **education** 11th, HS-grad, Assoc-acdm, Some-college, 10th, Prof-school, 7th-8th, Bachelors, Masters, Doctorate, 5th-6th, Assoc-voc, 9th, 12th, 1st-4th, Preschool;
3. **marital-status** Never-married, Married-civ-spouse, Widowed, Divorced, Separated, Married-spouse-absent, Married-AF-spouse;
4. **occupation** Machine-op-inspct, Farming-fishing, Protective-serv, Other-service, Prof-specialty, Craft-repair, Adm-clerical, Exec-managerial, Tech-support, Sales, Priv-house-serv, Transport-moving, Handlers-cleaners, Armed-Forces;
5. **relationship** Own-child, Husband, Not-in-family, Unmarried, Wife, Other-relative;
6. **race** Black, White, Asian-Pac-Islander, Other, Amer-Indian-Eskimo;
7. **sex** Male, Female;
8. **native-country** United-States, Cuba, Jamaica, India, Mexico, Puerto-Rico, Honduras, England, Canada, Germany, Iran, Philippines, Poland, Columbia, Cambodia, Thailand, Ecuador, Laos, Taiwan, Haiti, Portugal, Dominican-Republic, El-Salvador, France, Guatemala, Italy, China, South, Japan, Yugoslavia, Peru, Outlying-US(Guam-USVI-etc), Scotland, Trinidad&Tobago, Greece, Nicaragua, Vietnam, Hong, Ireland, Hungary, Holand-Netherlands.

参考文献

- [1] D. Dheeru and E. Karra Taniskidou, “UCI machine learning repository,” 2017.
- [2] “scikit-learn.” <http://scikit-learn.org/stable/>.
- [3] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.
- [4] “Cs231n: Convolutional neural networks for visual recognition.” <http://cs231n.github.io/neural-networks-3/>.
- [5] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [6] “Python.” <https://www.python.org/>.
- [7] “Jupyter notebook.” <http://jupyter.org/>.
- [8] “Numpy.” <http://www.numpy.org/>.
- [9] “Pandas.” <http://pandas.pydata.org/>.
- [10] “matplotlib.” <https://matplotlib.org/>.
- [11] “scipy.” <https://www.scipy.org/>.